Mid-Term_Project_What's_Cooking

Brooke Fitzgerald

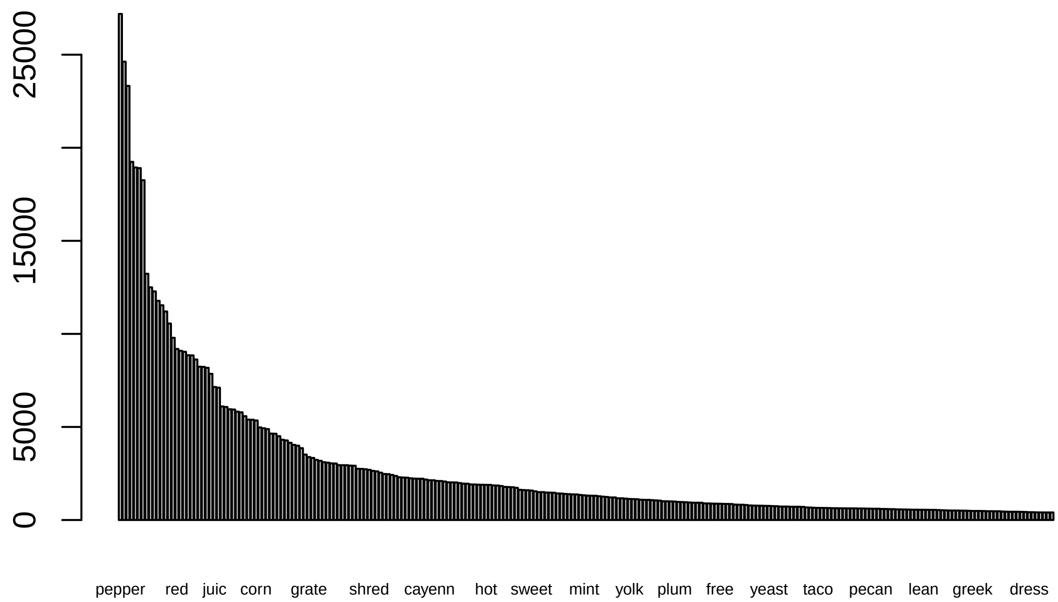
10/25/2015

For my midterm project I analyzed data from Kaggle's data competition "What's Cooking?" containing recipes, their ingredients, and their cuisines. First thing's first, I downloaded the training data set, transformed it into a document term matrix, and did some data pre-processing with the tm package. I removed the punctuation, ensured that all of the ingredients were lowercase, removed common English words, stemmed all of the words, and removed sparse terms.

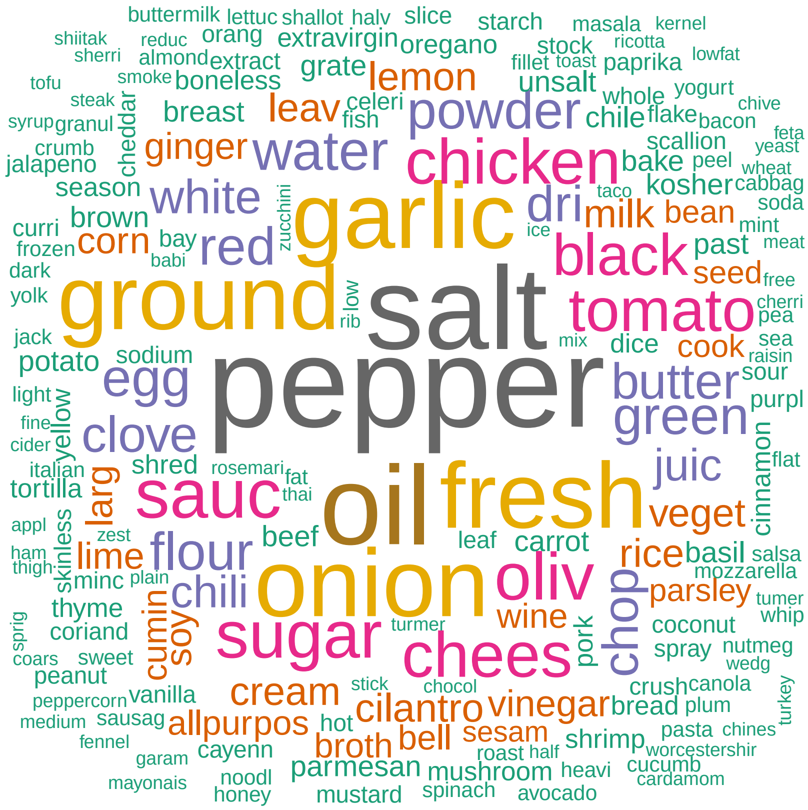
```
# textual preprocessing
ingredients <- Corpus(VectorSource(train$ingredients))
ingredients <- tm_map(ingredients, removePunctuation)
ingredients <- tm_map(ingredients, content_transformer(tolower))
ingredients <- tm_map(ingredients, removeWords, stopwords("english"))
ingredients <- tm_map(ingredients, stemDocument)

dtm <- DocumentTermMatrix(ingredients)
sparsedtm <- removeSparseTerms(dtm, 0.99)
ingredientsDTM <- as.data.frame(as.matrix(sparsedtm))
```

Then I wanted to examine the frequency and distribution of the different ingredients across the data set both with a bar plot and a word cloud of the ingredient frequencies. I also looked at the recipe distribution by cuisine.

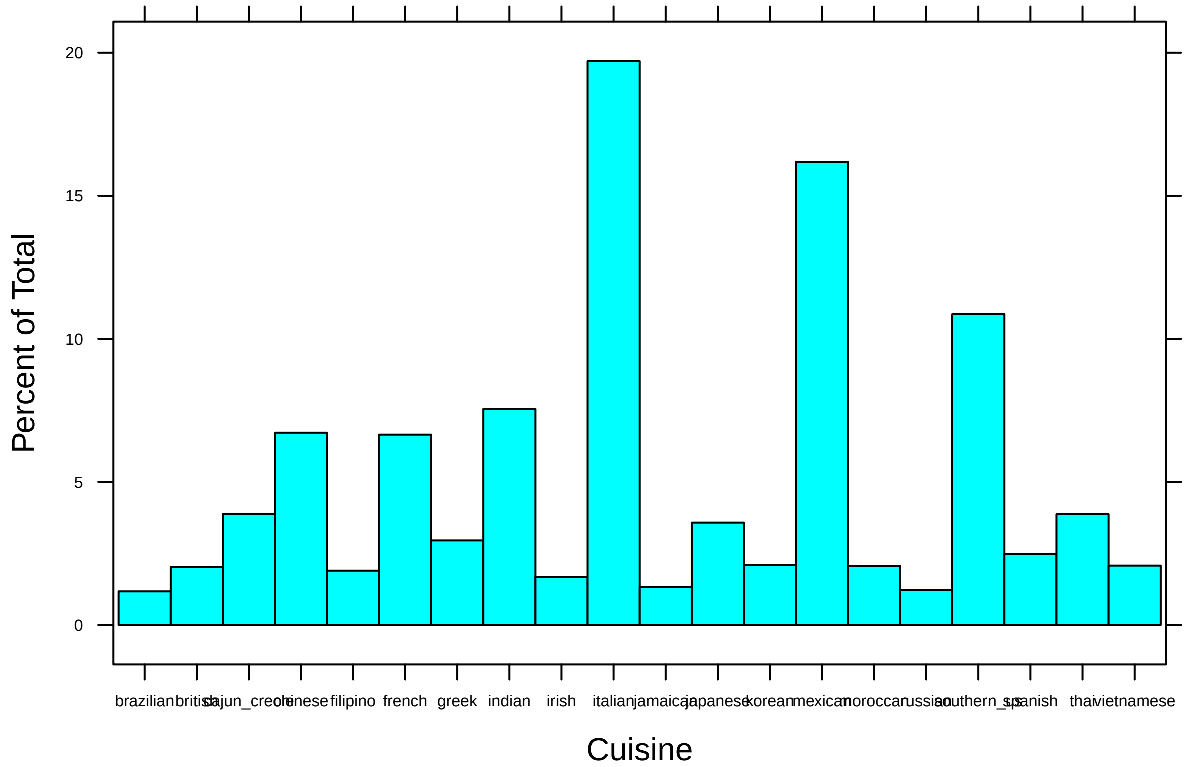
```
# Barplot
ingredientFreq <- sort(colSums(ingredientsDTM), decreasing = TRUE)
ingredientNames <- names(ingredientFreq)
barplot(ingredientFreq, cex.names = 0.5)
```




```r
# Wordcloud
wordcloud(ingredients, max.words = 200, random.order = FALSE, colors = brewer.pal(8,
    "Dark2"))
```




```r
# Add the dependent variable to the data.frame
ingredientsDTM$cuisine <- as.factor(train$cuisine)
histogram(ingredientsDTM$cuisine, scales = list(cex = 0.5), xlab = "Cuisine")
```



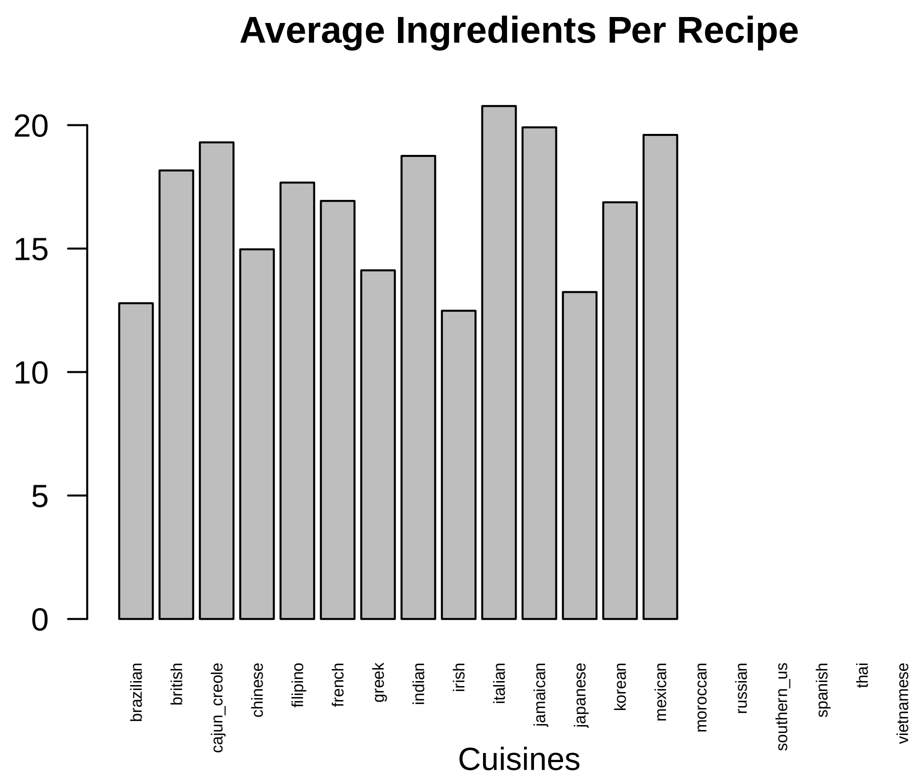
After seeing how the different ingredients were distributed, I decided to look more at cuisine and what it does to the data. I created another document term matrix that has less ingredients and is thus better for broad visualization. I then split that data by cuisine and manipulated the data to see what percent of the recipes for each cuisine contain a particular ingredient. I then added up all of those percents for each cuisine and got an estimate of how many ingredients are used in each cuisine's average recipes.

```
# remove some ingredients to have a more maneagable size for visualization
moreSparseDTM <- removeSparseTerms(dtm, 0.935)
smalldtm <- as.data.frame(as.matrix(moreSparseDTM))
smalldtm$cuisine <- as.factor(train$cuisine)
splitIngredients <- split(smalldtm, smalldtm$cuisine)
# since all of the cuisines have the same ingredients, deleting the last row
# of any cuisine will give you the ingredient names
ingredientNames <- names(splitIngredients$brazilian)
length(ingredientNames) <- (length(ingredientNames) - 1)

row.data <- c()

for (cuisine in splitIngredients) {
    # remove the cuisine column at the end
    woCuisine <- cuisine[, -(ncol(cuisine))]
    # add the column (ingredient) names back in
    names(woCuisine) <- ingredientNames
    # sum the column which will give you the number of times the ingredient was
    # used per cuisine
    ingPerCuisine <- colSums(woCuisine)
```

```r
    # dividing the columns by the number of recipes gives you an estimate of how
    # many ingredients are used in each cuisine's average recipes.
    ingPerCuisine <- ingPerCuisine/nrow(woCuisine)
    row.data <- c(row.data, ingPerCuisine)
}

ingPerCuisine.matrix <- rbind(row.data[1:90], row.data[91:180], row.data[181:270],
    row.data[271:360], row.data[361:450], row.data[451:540], row.data[541:630],
    row.data[631:720], row.data[721:810], row.data[811:900], row.data[901:990],
    row.data[991:1080], row.data[1081:1170], row.data[1171:1260], row.data[1261:1350],
    row.data[1351:1440], row.data[1441:1530], row.data[1531:1620], row.data[1621:1710],
    row.data[1711:1800])
rownames(ingPerCuisine.matrix) <- names(splitIngredients)
ingPerCuisine.matrix[1:3, 1:10]
```

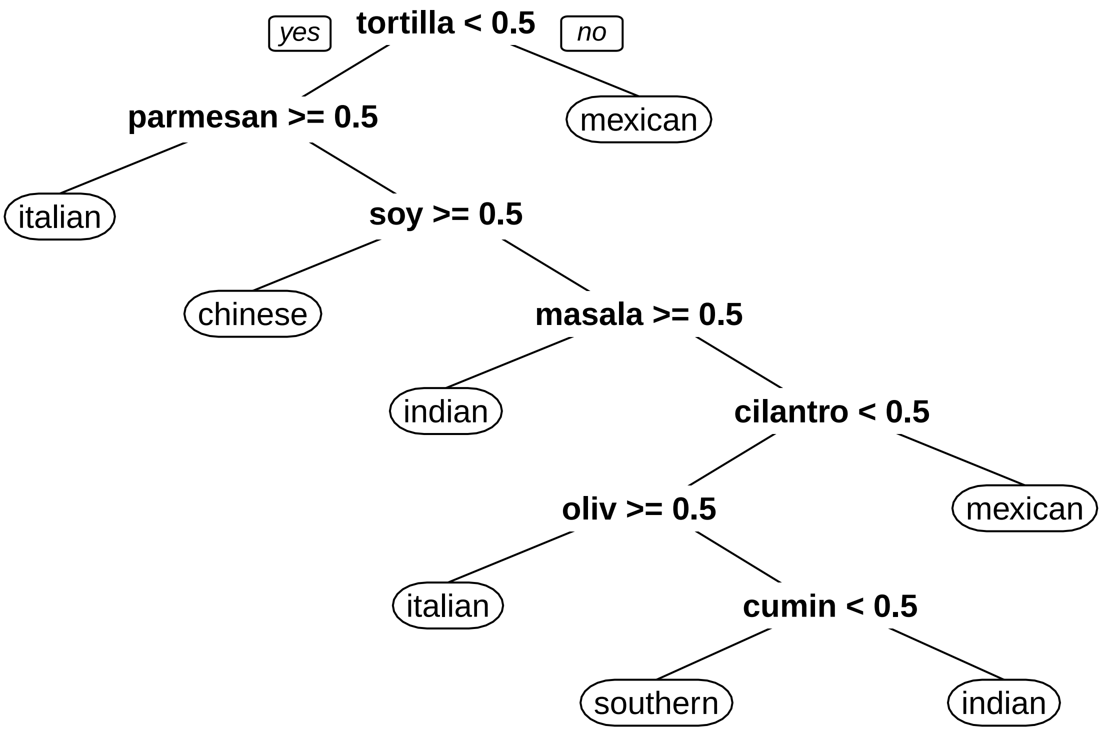
```
##                allpurpos       basil      bean       beef      bell
## brazilian    0.03854390 0.004282655 0.1520343 0.06638116 0.1713062
## british      0.54228856 0.023631841 0.5808458 0.22388060 0.1169154
## cajun_creole 0.03751617 0.291073739 0.3589909 0.34734799 0.6423027
##                   black   boneless       breast       broth      brown
## brazilian    0.28479657 0.03426124 0.0364025696 0.070663812 0.04496788
## british      0.06716418 0.06716418 0.0323383085 0.388059701 0.08084577
## cajun_creole 0.43014230 0.02199224 0.0006468305 0.009055627 0.20051746
```

```r
ingredientsPerRecipe <- rowSums(ingPerCuisine.matrix)
par(las = 2)
par(mar = c(5, 8, 4, 2))
barplot(ingredientsPerRecipe, xlab = "Cuisines", cex.names = 0.5, main = "Average Ingredients Per Recipe
```



After that, I wanted to create a decision tree that would predict the cuisine of a recipe based on it's ingredients. To do this, I split up my data into a training and test set, and made a model with the training data. I then created a rpart decision tree model and ran it on my test set data. I then qualified my results with a confusion matrix.

```
####### Decision Tree #######
inTrain <- createDataPartition(y = ingredientsDTM$cuisine, p = 0.6, list = FALSE)
training <- ingredientsDTM[inTrain, ]
testing <- ingredientsDTM[-inTrain, ]

treemodel <- rpart(cuisine ~ ., data = training, method = "class")
prp(treemodel)
```




```r
# Predict using the decision tree
prediction <- predict(treemodel, newdata = testing, type = "class")

# confusion matrix
CM <- confusionMatrix(prediction, testing$cuisine)
CM
```

```
## Confusion Matrix and Statistics
##
##                Reference
## Prediction      brazilian british cajun_creole chinese filipino french
##   brazilian             0       0            0       0        0      0
##   british               0       0            0       0        0      0
##   cajun_creole          0       0            0       0        0      0
##   chinese               1       3            6     823      120      2
##   filipino              0       0            0       0        0      0
##   french                0       0            0       0        0      0
##   greek                 0       0            0       0        0      0
##   indian                4       8           19       1        2      2
##   irish                 0       0            0       0        0      0
##   italian              47      29          169      12       13    323
##   jamaican              0       0            0       0        0      0
##   japanese              0       0            0       0        0      0
##   korean                0       0            0       0        0      0
##   mexican              35       2           12      28        4      8
##   moroccan              0       0            0       0        0      0
```

```
##   russian              0       0            0       0        0      0
##   southern_us         99     279          412     205      163    723
##   spanish              0       0            0       0        0      0
##   thai                 0       0            0       0        0      0
##   vietnamese           0       0            0       0        0      0
##               Reference
## Prediction     greek indian irish italian jamaican japanese korean mexican
##   brazilian        0      0     0       0        0        0      0       0
##   british          0      0     0       0        0        0      0       0
##   cajun_creole     0      0     0       0        0        0      0       0
##   chinese          3     16     1       7       38      300    206      10
##   filipino         0      0     0       0        0        0      0       0
##   french           0      0     0       0        0        0      0       0
##   greek            0      0     0       0        0        0      0       0
##   indian           7    554     2       3        7       28      0     181
##   irish            0      0     0       0        0        0      0       0
##   italian        315     72    24    2202       20       14      3     220
##   jamaican         0      0     0       0        0        0      0       0
##   japanese         0      0     0       0        0        0      0       0
##   korean           0      0     0       0        0        0      0       0
##   mexican          4    239     5      25       16       10     10    1588
##   moroccan         0      0     0       0        0        0      0       0
##   russian          0      0     0       0        0        0      0       0
##   southern_us    141    320   234     898      129      217    113     576
##   spanish          0      0     0       0        0        0      0       0
##   thai             0      0     0       0        0        0      0       0
##   vietnamese       0      0     0       0        0        0      0       0
##               Reference
## Prediction     moroccan russian southern_us spanish thai vietnamese
##   brazilian           0       0           0       0    0          0
##   british             0       0           0       0    0          0
##   cajun_creole        0       0           0       0    0          0
##   chinese             1       2          14       2  206         85
##   filipino            0       0           0       0    0          0
##   french              0       0           0       0    0          0
##   greek               0       0           0       0    0          0
##   indian             32       1          18       2   11          1
##   irish               0       0           0       0    0          0
##   italian           127      26         197     227   14          8
##   jamaican            0       0           0       0    0          0
##   japanese            0       0           0       0    0          0
##   korean              0       0           0       0    0          0
##   mexican           114       2          23      40  197         94
##   moroccan            0       0           0       0    0          0
##   russian             0       0           0       0    0          0
##   southern_us        54     164        1476     124  187        142
##   spanish             0       0           0       0    0          0
##   thai                0       0           0       0    0          0
##   vietnamese          0       0           0       0    0          0
##
## Overall Statistics
##
##                Accuracy : 0.4177
##                  95% CI : (0.41, 0.4254)
```

```
##     No Information Rate : 0.1971
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.3285
##  Mcnemar's Test P-Value : NA
##
## Statistics by Class:
##
##                      Class: brazilian Class: british Class: cajun_creole
## Sensitivity                    0.0000        0.00000             0.00000
## Specificity                    1.0000        1.00000             1.00000
## Pos Pred Value                    NaN            NaN                 NaN
## Neg Pred Value                 0.9883        0.97982             0.96114
## Prevalence                     0.0117        0.02018             0.03886
## Detection Rate                 0.0000        0.00000             0.00000
## Detection Prevalence           0.0000        0.00000             0.00000
## Balanced Accuracy              0.5000        0.50000             0.50000
##                      Class: chinese Class: filipino Class: french
## Sensitivity                 0.76988         0.00000       0.00000
## Specificity                 0.93104         1.00000       1.00000
## Pos Pred Value              0.44583             NaN           NaN
## Neg Pred Value              0.98250         0.98101       0.93347
## Prevalence                  0.06722         0.01899       0.06653
## Detection Rate              0.05175         0.00000       0.00000
## Detection Prevalence        0.11608         0.00000       0.00000
## Balanced Accuracy           0.85046         0.50000       0.50000
##                      Class: greek Class: indian Class: irish
## Sensitivity               0.00000       0.46128      0.00000
## Specificity               1.00000       0.97762      1.00000
## Pos Pred Value                NaN       0.62741          NaN
## Neg Pred Value            0.97045       0.95692      0.98327
## Prevalence                0.02955       0.07552      0.01673
## Detection Rate            0.00000       0.03484      0.00000
## Detection Prevalence      0.00000       0.05552      0.00000
## Balanced Accuracy         0.50000       0.71945      0.50000
##                      Class: italian Class: jamaican Class: japanese
## Sensitivity                  0.7024         0.00000         0.00000
## Specificity                  0.8543         1.00000         1.00000
## Pos Pred Value               0.5421             NaN             NaN
## Neg Pred Value               0.9212         0.98679         0.96422
## Prevalence                   0.1971         0.01321         0.03578
## Detection Rate               0.1385         0.00000         0.00000
## Detection Prevalence         0.2554         0.00000         0.00000
## Balanced Accuracy            0.7784         0.50000         0.50000
##                      Class: korean Class: mexican Class: moroccan
## Sensitivity                0.00000        0.61670         0.00000
## Specificity                1.00000        0.93487         1.00000
## Pos Pred Value                 NaN        0.64658             NaN
## Neg Pred Value             0.97912        0.92660         0.97937
## Prevalence                 0.02088        0.16192         0.02063
## Detection Rate             0.00000        0.09986         0.00000
## Detection Prevalence       0.00000        0.15444         0.00000
## Balanced Accuracy          0.50000        0.77579         0.50000
##                      Class: russian Class: southern_us Class: spanish
```

```
## Sensitivity                 0.00000            0.85417        0.00000
## Specificity                 1.00000            0.63457        1.00000
## Pos Pred Value                  NaN            0.22175            NaN
## Neg Pred Value              0.98774            0.97275        0.97516
## Prevalence                  0.01226            0.10866        0.02484
## Detection Rate              0.00000            0.09281        0.00000
## Detection Prevalence        0.00000            0.41854        0.00000
## Balanced Accuracy           0.50000            0.74437        0.50000
##                      Class: thai Class: vietnamese
## Sensitivity              0.00000           0.00000
## Specificity              1.00000           1.00000
## Pos Pred Value               NaN               NaN
## Neg Pred Value           0.96133           0.97925
## Prevalence               0.03867           0.02075
## Detection Rate           0.00000           0.00000
## Detection Prevalence     0.00000           0.00000
## Balanced Accuracy        0.50000           0.50000
```

As you can see, my accuracy rate is fairly low at 40% overall accuracy. However, my model preforms 8 times better than pure chance and a little over twice as well as just predicting every recipe as Italian.